

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2002-64713

(P2002-64713A)

(43) 公開日 平成14年2月28日 (2002.2.28)

| (51) Int.Cl. <sup>7</sup> | 識別記号 | F I           | テマコード* (参考) |
|---------------------------|------|---------------|-------------|
| H 0 4 N 1/41              |      | H 0 4 N 1/41  | Z 5 C 0 5 9 |
| H 0 3 M 7/30              |      | H 0 3 M 7/30  | B 5 C 0 7 8 |
| H 0 4 N 7/32              |      | H 0 4 N 7/137 | Z 5 J 0 6 4 |

審査請求 未請求 請求項の数10 O L (全 11 頁)

(21) 出願番号 特願2000-250287(P2000-250287)

(22) 出願日 平成12年8月21日 (2000.8.21)

(71) 出願人 598158521

アイ・アンド・エフ株式会社

東京都文京区本郷4丁目1番4号 コスモ  
ス本郷ビル

(72) 発明者 大見 忠弘

宮城県仙台市青葉区米ヶ袋2-1-17-  
301

(72) 発明者 中山 貴裕

宮城県仙台市青葉区荒巻字青葉 (無番地)  
東北大学内

(74) 代理人 100090273

弁理士 國分 孝悦

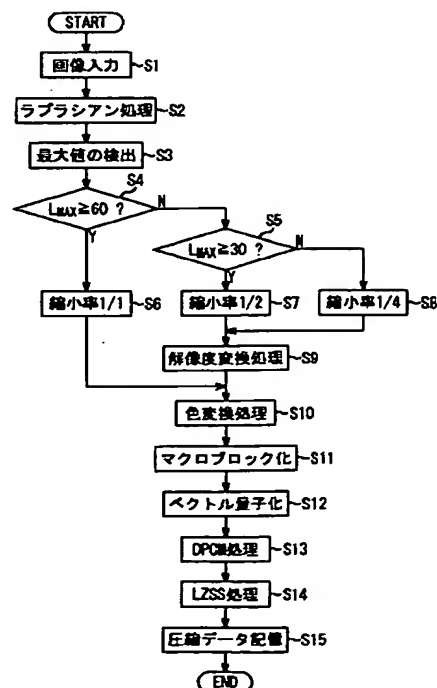
最終頁に続く

(54) 【発明の名称】 データ処理方法、データ処理装置及び記憶媒体

(57) 【要約】

【課題】 コントラストが低い部分がある画像に対して、再生画像の画質を維持したまま更なる圧縮率を向上させた画像圧縮を行う。

【解決手段】 入力画像データの所定領域において、R G B信号毎にラプラシアンフィルタ処理を行う第1のステップと、R G B信号毎のラプラシアンフィルタ処理後の値から最大値を抽出する第2のステップと、最大値の大きさに応じて前記所定領域の解像度変換処理を行う第3のステップとを有する。R G B信号毎にラプラシアンフィルタ処理を行い、最大値を検出することにより、コントラストが低い部分がある画像であっても、画像のエッジを確実に検出してエッジ量に応じた圧縮を行うことが可能となり、再生画像の画質を維持したまま、更なる圧縮率の向上を達成することができる。



## 【特許請求の範囲】

【請求項1】 入力画像データの所定領域において、RGB信号毎にラプラシアンフィルタ処理を行う第1のステップと、

前記RGB信号毎のラプラシアンフィルタ処理後の値から最大値を抽出する第2のステップと、

前記最大値の大きさに応じて前記所定領域の解像度変換処理を行う第3のステップとを有することを特徴とするデータ処理方法。

【請求項2】 前記第3のステップの後、前記所定領域をマクロブロックに分割してベクトル量子化を行う第4のステップと、

前記ベクトル量子化後の出力データに対して差分予測符号化処理を行う第5のステップと、

前記差分予測符号化処理後の出力データに対してLZSS辞書圧縮を行う第6のステップとを更に有することを特徴とする請求項1に記載のデータ処理方法。

【請求項3】 前記最大値の大きさに応じて前記解像度変換処理における縮小率を変更することを特徴とする請求項1又は2に記載のデータ処理方法。

【請求項4】 前記第3のステップにおいて、前記最大値と予め設定しておいた少なくとも1つのしきい値とを比較して、前記縮小率を変更することを特徴とする請求項3に記載のデータ処理方法。

【請求項5】 複数の画素から成る入力画像データの所定領域において、画像のエッジ検出を行うエッジ検出手段と、

前記所定領域における前記画像のエッジの量に応じて前記所定領域内の前記画素の信号を解像度変換処理してデータ圧縮を行う圧縮手段とを有することを特徴とするデータ処理装置。

【請求項6】 前記圧縮手段によりデータ圧縮された前記所定領域をマクロブロックに分割してベクトル量子化を行うベクトル量子化手段と、

前記ベクトル量子化手段から出力された出力データ相互の差分を演算してデータ量を圧縮する差分予測符号化手段と、

前記差分予測符号化手段から出力された圧縮されたデータ列に対してLZSS辞書圧縮を行うLZSS圧縮手段を更に備えたことを特徴とする請求項5に記載のデータ処理装置。

【請求項7】 前記エッジ検出手段は、前記所定領域においてそれぞれの前記画素のRGB信号毎にラプラシアンフィルタ処理を行うラプラシアンフィルタ処理手段と、

前記ラプラシアンフィルタ処理後の値から最大値を抽出する最大値抽出手段と、

前記最大値と所定のしきい値とを比較する比較手段とを有することを特徴とする請求項5又は6に記載のデータ処理装置。

【請求項8】 前記圧縮手段は、前記比較手段による比較の結果に応じて前記解像度変換処理の縮小率を決定することを特徴とする請求項5～7のいずれか1項に記載のデータ処理装置。

【請求項9】 請求項1～4のいずれか1項に記載のデータ処理方法の手順をコンピュータに実行させるためのプログラムを記憶したコンピュータ読み取り可能な記憶媒体。

【請求項10】 請求項5～8のいずれか1項に記載のデータ処理装置の各手段としてコンピュータに機能させるためのプログラムを記憶したコンピュータ読み取り可能な記憶媒体。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、データ処理装置及び処理方法及びこれらの処理を実行させるためのプログラムを記憶した記憶媒体に関し、特に、データ圧縮手法の1つとしてベクトル量子化を用いる処理装置、処理方法及び記憶媒体に用いて好適なものである。

## 【0002】

【従来の技術】従来、データ圧縮の手法が種々提案されている。その中で、圧縮データの伸長処理を非常に簡単に行うことが可能なデータ圧縮アルゴリズムの1つとして、「ベクトル量子化」という手法が良く用いられている。このアルゴリズムは、古くから信号処理の分野で知られており、特に、画像信号や音声信号のデータ圧縮、あるいはパターン認識に応用されてきた。

【0003】このベクトル量子化では、ある大きさ（例えば4×4画素のブロック）の画素パターンを幾つか用意しておき、それぞれにユニークな番号などを与える（この集合体を「コードブック」という）。そして、例えば2次元配列の画像データ中から同じ大きさ（例えば4×4画素）のブロックを順次取り出し、それと最も似通ったパターンをコードブック中から見つけ出して、そのパターンの番号を当該ブロックに当てはめるというデータ圧縮を行う。ベクトル量子化では、1つのブロック内のデータ列が1つのコードベクトルに対応する。

【0004】このようにコード化された圧縮データの受信側あるいは伸長側では、各ブロック毎に番号に対応するパターンをコードブックの中から取り出すだけで、元の画像を再現することができる。従って、伸長側では、コードブックさえ受け取っているか、あるいは予め保持していれば、特に特殊な演算は必要としないため、非常に簡単なハードウェアで元の画像を再生することが可能となる。

## 【0005】

【発明が解決しようとする課題】しかしながら、画像データ中の特徴量を示す各パラメータは、通常、領域毎にバラツキの度合いが異なるものである。例えば輝度を例にとると、1画像データ中にはエッジ成分が多く存在

し、輝度変化が大きい領域と、背景画像のように比較的輝度変化が少ない領域が存在する。そして、輝度変化の大きい領域の画像を圧縮する場合には、輝度変化が少ない領域よりも多くのコードブックのデータが必要になるため、輝度変化が少ない領域を含む画像の圧縮率を向上させることは困難であった。

【0006】また、輝度を予め検出して、輝度値に応じてベクトル量子化に用いるマクロブロックの大きさ、数を可変してデータ圧縮をしようとすると、例えば背景画像に対してコントラストの低い色文字等の情報は輝度値が低いものとして認識されてしまう。このため、輝度値に基づいて画像の各領域の圧縮率を可変させても良好な再生画像を得ることが困難であった。

【0007】本発明は、このような問題を解決するために成されたものであり、再生画像の画質を維持したまま、更なる圧縮率の向上を達成することを目的とする。

【0008】

【課題を解決するための手段】本発明のデータ処理方法は、入力画像データの所定領域において、RGB信号毎にラプラシアンフィルタ処理を行う第1のステップと、前記RGB信号毎のラプラシアンフィルタ処理後の値から最大値を抽出する第2のステップと、前記最大値の大きさに応じて前記所定領域の解像度変換処理を行う第3のステップとを有する。

【0009】本発明のデータ処理方法の一態様例においては、前記第3のステップの後、前記所定領域をマクロブロックに分割してベクトル量子化を行う第4のステップと、前記ベクトル量子化後の出力データに対して差分予測符号化処理を行う第5のステップと、前記差分予測符号化処理後の出力データに対してLZSS辞書圧縮を行う第6のステップとを更に有する。

【0010】本発明のデータ処理方法の一態様例においては、前記最大値の大きさに応じて前記解像度変換処理における縮小率を変更する。

【0011】本発明のデータ処理方法の一態様例においては、前記第3のステップにおいて、前記最大値と予め設定しておいた少なくとも1つのしきい値とを比較して、前記縮小率を変更する。

【0012】本発明のデータ処理装置は、複数の画素から成る入力画像データの所定領域において、画像のエッジ検出を行うエッジ検出手段と、前記所定領域における前記画像のエッジの量に応じて前記所定領域内の前記画素の信号を解像度変換処理してデータ圧縮を行う圧縮手段とを有する。

【0013】本発明のデータ処理装置の一態様例においては、前記圧縮手段によりデータ圧縮された前記所定領域をマクロブロックに分割してベクトル量子化を行うベクトル量子化手段と、前記ベクトル量子化手段から出力された出力データ相互の差分を演算してデータ量を圧縮する差分予測符号化手段と、前記差分予測符号化手段か

ら出力された圧縮されたデータ列に対してLZSS辞書圧縮を行うLZSS圧縮手段を更に備える。

【0014】本発明のデータ処理装置の一態様例においては、前記エッジ検出手段は、前記所定領域においてそれぞれの前記画素のRGB信号毎にラプラシアンフィルタ処理を行うラプラシアンフィルタ処理手段と、前記ラプラシアンフィルタ処理後の値から最大値を抽出する最大値抽出手段と、前記最大値と所定のしきい値とを比較する比較手段とを有する。

【0015】本発明のデータ処理装置の一態様例においては、前記圧縮手段は、前記比較手段による比較の結果に応じて前記解像度変換処理の縮小率を決定する。

【0016】本発明の記憶媒体は、上記のデータ処理方法の手順をコンピュータに実行させるためのプログラムを記憶したコンピュータ読み取り可能な記憶媒体である。

【0017】本発明の記憶媒体は、上記のデータ処理装置の各手段としてコンピュータに機能させるためのプログラムを記憶したコンピュータ読み取り可能な記憶媒体である。

【0018】

【発明の実施の形態】以下、本発明の一実施形態について図面を参照しながら詳細に説明する。以下に示す実施形態は、圧縮対象のデータとして例えば動画を構成する静止画を例にとり、画像データの圧縮率及び再生画質の更なる向上を達成するものである。

【0019】最初に、ベクトル量子化(VQ)による画像圧縮及び伸長の原理について、静止画像を例にとり図11を参照しながら説明する。図11に示すように、圧縮対象のデータとして入力される原画像1は、画素と呼ばれる要素が多数集まって構成されている。個々の画素は、RGB信号から変換された輝度信号(Y信号)、及び色信号(U、V信号)等の情報を持っている。

【0020】入力画像1中から複数の画素で構成されるブロックを取り出したのが、入力画像ブロック(マクロブロック)2である。図11の例では、入力画像ブロック2の大きさとして4×4画素を選んでいるが、この大きさは何であってもよい。この入力画像ブロック2は、上述の通り複数の画素を持っているから、各々の画素が持つ輝度信号値や色信号値を集めてそれぞれベクトルデータとすることができる。これが入力ベクトルデータである。

【0021】人間の視覚特性上、入力画像1中の幾つかの入力画像ブロック2は、見た目では殆ど同じに見える場合がある。こういった同じに見える複数の入力画像ブロック2を、より少ない数の画像ブロックで代表させることが可能である。画像ブロックコードブック3は、入力画像1上の多数の入力画像ブロック2を代表する画像ブロック(コードベクトルデータ)を複数持ったものである。コードベクトルデータは、画像ブロックコードブ

10

20

30

40

50

ック3内の画像ブロック各々の画素が持つ輝度信号値や色度信号値をベクトルデータとしたものである。

【0022】ベクトル量子化では、入力画像1の全体を画像ブロックとして分割し、各々の画像ブロック2を入力ベクトルデータとして、その入力ベクトルデータに類似するコードベクトルデータをコードブック3内から検索する。そして、該当するコードベクトルデータの番号のみを転送することで、画像を圧縮することが可能である。圧縮された画像を再生して再現画像4を得るには、上記転送された番号に対応するコードベクトルデータをコードブック3から読み出し、画像に当てはめればよい。

【0023】そして、本実施形態は、入力画像1の所定領域においてベクトル量子化を行う前にエッジ検出処理を行い、エッジ検出処理の結果に応じて所定領域において補間処理を行うことにより、ベクトル量子化前のデータ量を削減するようにしている。

【0024】例えば、背景画像に対して画像のエッジが多く検出される領域では、このエッジを高精度に再現するために補間処理は行わずにベクトル量子化を行う。一方で、エッジが殆ど検出されない領域においては、単調なパターンであることが想定され、補間処理により領域を縮小した後にベクトル量子化を行っても再生画像の画質の劣化は少ない。

【0025】従って、エッジが多く検出された領域のみ原画像のままベクトル量子化を行い、他の領域では補間処理により原画像のデータを縮小してベクトル量子化を行うことにより、全ての入力画像ブロック2を細かいマクロブロックによってベクトル量子化を行った場合と比較して圧縮率を大幅に向上させることが可能となる。

【0026】図1を参照しながら、入力画像の所定領域におけるエッジ検出の方法について説明する。本実施形態では $16 \times 16$ 画素の領域毎にエッジ検出を行い、その結果に応じて $16 \times 16$ 画素の領域を線形補間するようにしている。まず、図1(a)に示すように、所定領域5における各画素のRGB(R:赤、G:緑、B:青)信号に対して個別にラプラシアン処理を行い、RGB信号の各々から高周波成分を抽出する。これにより、各色信号に対してラプラシアン処理後のデータ $L_R$ 、 $L_G$ 、 $L_B$ がそれぞれ得られる(図1(b)参照)。

【0027】図1において、6はラプラシアンフィルタの一例を示している。ラプラシアンフィルタ6は、中央部とその周辺部に重みを設定したフィルタであり、重み値を全て合計すると“0”となるフィルタである。この例では、中央部の重みを“-4”、中央部に対してY軸方向、及びX軸方向の重みをそれぞれ“1”に設定している。そして、中央部に対して斜め方向の重みは“0”に設定している。

【0028】原画像データに対してラプラシアンフィルタ6をかけると、原画像データの各画素のRGBデータ

が“0”に平均化され、値がプラス方向とマイナス方向とに分布するエッジ成分だけが残る。すなわち、ラプラシアンフィルタ6をかけることによって、変化量のみで表された高周波成分のデータが各RGB信号から抽出される。

【0029】このように、RGBの各信号に対してラプラシアンフィルタ6をかけることにより、特に所定領域5におけるエッジ部分のコントラストが低い場合であっても、確実にエッジ検出を行うことが可能となる。例えば、背景画が白色で、黄色の文字が描かれているような場合には、輝度変換を行った後にエッジ検出を行うと、黄色と白色のコントラストが低いため、エッジ検出が困難であったが、本実施形態のように輝度変換を行う前に各RGB信号をラプラシアン処理することによって、輝度のコントラストが低い場合であっても、確実にエッジ検出を行うことが可能となる。

【0030】所定領域5においてラプラシアン処理を行った後、所定領域5内の同一箇所に相当するそれぞれの画素においてRGB信号をラプラシアン処理して得られた値 $L_R$ 、 $L_G$ 、 $L_B$ のうちの最大値を求める。図1の例では、図1(b)に示すように同一画素における各RGB信号をラプラシアン処理して得られた値 $L_{R1}$ 、 $L_{G1}$ 、 $L_{B1}$ のうちから最大値を抽出し、 $16 \times 16$ 画素の所定領域5の全ての画素において最大値をそれぞれ求める。そして、所定領域5の各々の画素における最大値の集合としてのデータ群が $L_{MAX}$ が得られ(図1(c)参照)。

【0031】そして、データ群 $L_{MAX}$ のうち、少なくとも1つの画素の最大値が所定のしきい値を超えている場合には、所定領域5中にエッジ成分が存在するものと判断する。

【0032】このように、ラプラシアン処理後のRGB信号の各値から、最大値を抽出し、所定領域5の全画素における最大値のデータ群( $L_{MAX}$ )と所定のしきい値を比較してエッジの有無を判定することにより、例えば、R信号からはエッジが検出されないような画像であっても、他のG、B信号からエッジを検出することが可能となり、所定領域5中のエッジの検出漏れを最小限に抑えることが可能となる。

【0033】次に、図2及び図3を参照しながら、各所定領域5におけるエッジ検出の結果に基づいて解像度変換処理を行ってデータ圧縮を行う処理について説明する。本実施形態では、各所定領域5中におけるエッジ判定の結果得られたデータ群 $L_{MAX}$ の値と、予め定めておいた所定のしきい値との比較を行い、比較の結果に基づいて、解像度変換処理を行うようにしている。ここで、比較の際に用いるしきい値としては30、60の2つの値を用いている。

【0034】図2(b)に示すように、ある所定領域5において得られたエッジ値( $L_{MAX}$ )の値が60以上で

あった場合には、その所定領域5においてはエッジ成分が多いことが判別できるため、解像度変換処理は行わずに通常通り4×4のマクロブロックを16個用いて所定領域5のベクトル量子化を行う。

【0035】一方、ある所定領域5において得られたエッジ値(L<sub>MAX</sub>)の値が30未満であった場合には、その所定領域5においてはエッジ成分が少ないことが判別できるため、縮小率を1/4として所定領域5の解像度

$$s(x+p, y+q) = (1-q) \times ((1-p) \times s(x, y) + p \times s(x+1, y)) + q \times ((1-p) \times s(x, y+1) + p \times s(x+1, y+1))$$

【0037】ここで、p、qは0から1の間の実数である。

【0038】図2(a)は、1例として縮小率を1/4とした場合の解像度変換処理を示している。ここでは、4つの画素11~14が解像度変換処理により1つの画素15に代表されている。これにより、解像度を1/4にし、ベクトル量子化の対象となるマクロブロック数を削減することが可能となる。

【0039】すなわち、16×16画素の所定領域5においては、4×4のマクロブロックを16個用いてベクトル量子化を行うが、縮小率を1/4としてを行っているため4×4のマクロブロックを4個用いるだけで所定領域5の画像を代表させ、その後のベクトル量子化を行うことができる。これにより、L<sub>MAX</sub>の値が30未満の所定領域5においては圧縮率を4倍に高めることができる。

【0040】また、ある所定領域5において得られたエッジ値(L<sub>MAX</sub>)の値が30以上60未満であった場合には、上述の2つの場合の中間の処理を行う。すなわち、この場合には縮小率を1/2に設定して解像度変換処理を行い、ここでは、4×4のマクロブロックを8個用いて、その後のベクトル量子化処理を行う。これにより、L<sub>MAX</sub>の値が30以上60未満の所定領域5においては圧縮率を2倍まで高めることができる。

【0041】図3は、上述の解像度変換処理により所定領域5のデータ量を削減する方法を模式的に示した図である。

【0042】図3(a)は、エッジ判定の結果、所定領域5中にエッジ成分が多く検出され、L<sub>MAX</sub>のうち少なくとも1つの値が60以上となった場合の処理を示している。この場合、エッジ成分を精度よく再現する必要があるため、所定領域5における補間処理は行わずに4×4のマクロブロックを16個用いてベクトル量子化を行う。

【0043】図3(b)は、所定領域5中にエッジ成分が多く検出され、L<sub>MAX</sub>のうち少なくとも1つの値が30以上60未満となった場合の処理を示している。この場合、エッジ成分が比較的少ないため、図3(a)の場合ほど細かいマクロブロックに分割する必要はない。従って、縮小率を1/2とし、16×16画素の所定領域

変換処理を行いベクトル量子化の前段階で解像度変換を行う。そして、データ圧縮後、ベクトル量子化のためのマクロブロック化を行う。解像度変換処理の一例として、本実施形態では線形補間縮小処理を用いており、例えば、以下の式により行う。

【0036】

【数1】

5を解像度変換処理により16×8画素として、4×4のマクロブロックを8個用いてベクトル量子化を行う。

【0044】図3(c)は、所定領域5中にエッジ成分が多く検出され、L<sub>MAX</sub>の値が30未満となった場合の処理を示している。この場合、エッジ成分はほとんど検出されず、いわゆるベタの画像となるため、図2(a)で説明したように縮小率を1/4とし、16×16画素の所定領域5を解像度変換処理により8×8画素として、4×4のマクロブロックを4個用いたベクトル量子化を行う。従って、この場合圧縮率を4倍まで高めることが可能となる。

【0045】このように、検出したエッジの量に基づいて所定領域5の解像度変換処理を行い、ベクトル量子化に用いるマクロブロックの個数を可変した後ベクトル量子化を行う。そして、ベクトル量子化の結果、コードブック3から検索されたコードベクトルデータに従ったコード番号が出力される。

【0046】次に、図4を参照しながら、ベクトル量子化により出力されたコード番号のデータを更に圧縮する方法について説明する。本実施形態では、コード番号データを更に圧縮するために、差分予測符号化(DPCM)処理とLZSS辞書圧縮処理という2つの処理を行っている。LZSS辞書圧縮処理は、特にベクトル量子化後の圧縮データ列に対して親和性の高い圧縮処理であるが、本実施形態では、LZSS辞書圧縮処理の前段に差分予測符号化処理を行い、得られたデータ列に対してLZSS辞書圧縮処理を行うことにより、LZSS辞書圧縮処理の効果を更に高めるようにしている。

【0047】ベクトル量子化後のコード番号列は、隣合うマクロブロックのコード番号が順次に出力されているため、隣同士のコード番号の相関が高く、差分予測符号化処理を行うことにより大幅にデータ数を削減することができる。

【0048】図4を参照しながら、差分予測符号化処理について詳細に説明する。図4は、ベクトル量子化後の隣接するマクロブロックの出力コードの差の分布を示している。図4に示すように、ベクトル量子化後の隣接するコード番号の出力コードの差の約90%は±119以内となっている。上述したように、通常、隣接するマクロブロックにおいては画像が急激に変化することは稀で

あるため、出力コードの相関が非常に高いためである。

【0049】差分予測符号化処理では、このことを利用して、差が±119の範囲内のデータについては新たな出力コードを与えて8ビットとするようにしている。これにより、差が±119以内のコード番号列についてはベクトル量子化の直後では出力ビット数が18ビットであるのに対し、8ビットまでデータ圧縮を行うことが可能となる。

【0050】図5は、差分予測符号化処理により得られるコード列を示している。図5に示すように、差が+119〜-119の範囲内であるときは、ベクトル量子化により出力されたコード番号の代わりに差のコードを用いる。例えば、差の値が-1の場合には出力コードを“1”に、差の値が+5の場合には出力コードを“10”とする。これにより、ベクトル量子化により出力された16ビットのデータを8ビットのデータに圧縮することが可能となる。なお、差の値が±119よりも大きい場合には、0×F000とコード番号との論理和(OR)を取った16ビットデータを出力する。

【0051】8ビットに圧縮したデータは、特にLZSS辞書圧縮との親和性が高いため、その後のLZSS圧縮の圧縮の効果をより向上させることができる。また、16ビットのままのデータについても、8ビットのデータ列2つ分に相当するため、LZSS圧縮の親和性を維持することができる。

【0052】図6は、ベクトル量子化により出力されたコード番号、差の値、DPCM処理後のデータの関係を模式的に示した図である。図6に示すように、ベクトル量子化による出力データが“2”，“3”，“1024”，“1023”，“5”…の場合、隣り合うコード番号の差は、“2”と“3”の間では $2-3=-1$ 、“3”と“1024”の間では $3-1024=-1021$ 、“1024”と“1023”の間では $1024-1023=+1$ 、“1023”と“5”の間では $1023-5=+1018$ となる。

【0053】そして、差の値が-1，+1の場合には、±119の範囲内であるため、元の出力コード“3”，“1023”を用いず、差の値-1，+1に置き換える。これにより、置き換えたデータのビット数を16ビットから8ビットにデータ圧縮することができる。なお、差分予測符号化処理において、差の値のしきい値を±119とすると8ビットへのデータ圧縮が可能であるが、例えば4ビットへのデータ圧縮を行う場合には、差の値が±7以内の出力コードについてデータ変換を行うようにする。これにより、更なるデータ圧縮を達成することが可能となる。

【0054】そして、本実施形態では、差分予測符号化処理後のデータの集合に対してLZSS辞書圧縮を行うことにより、更なるデータの圧縮を達成している。後述するように、LZSS辞書圧縮はベクトル量子化後のデ

ータ列と特に親和性が高いため、少なくともベクトル量子化後のデータ列に対してLZSS辞書圧縮を行うことにより、データの更なる圧縮を達成することができる。すなわちLZSS辞書圧縮は、ベクトル量子化後のデータを更に圧縮する方法であり、ベクトル量子化圧縮後のコード番号列に対して、LZSS辞書圧縮を行い、更なるデータ圧縮をかける。LZSS辞書圧縮は完全可逆圧縮であるので、再生時にはLZSS辞書圧縮前のベクトル量子化によるコード番号列に戻すことができる。

【0055】ここで、LZSS辞書圧縮とは、過去のデータを辞書として参照し、その辞書内の最長一致データ列を用いて符号化するものであり、同じ文字、記号等を繰り返し用いているデータに対して圧縮効果が大きくなる。図9を用いて、LZSS辞書圧縮について簡単に説明する。図9(a)はLZSS辞書圧縮前のデータ列を示している。LZSS辞書圧縮前のデータ列では、2番目から6番目までのデータ列(BCDEF)と、7番目から11番目までのデータ列(BCDEF)が同一のデータとされている。

【0056】LZSS辞書圧縮は、先に表れたデータ列と同一のデータ列を、先のデータ列を辞書として利用して表記する方法であり、この例では7番目から11番目までのデータ列(BCDEF)を、先の2番目から6番目までのデータ列(BCDEF)を辞書として利用して表記する。

【0057】図9(b)がLZSS辞書圧縮後のデータ列を示している。7番目から11番目までのデータ列(BCDEF)は、2番目から5個連続するデータ列と同じであるため、(2, 5)と表記される。通常、1つのデータ(例えば1番目のA)は1バイトで構成されるが、図9(b)の(2, 5)のデータ量は2バイトである。従って、7番目から11番目までのデータ列(BCDEF)は、本来5バイトのデータが必要とされていたが2バイトで済むこととなり、3バイト分のデータ圧縮を行うことが可能となる。

【0058】図9(c)は、(2, 5)を記憶する2バイトのデータ列を示している。ここで、辞書として参照したデータ列の先頭位置を示す“2”については12ビット長のデータで表され、その辞書に含まれるデータ数を示す“5”については4ビット長のデータで表される。このように辞書内の最長一致データの数を表すデータを4ビット長とした場合は、連続するデータの数として通常は0〜15個までのデータ列を辞書として記憶することができるが、4ビット長のデータ値に所定値を加算して表すことにより、例えば1〜16個、4〜19個などより多くのデータ列を辞書として記憶させることも可能である。

【0059】一方、ベクトル量子化は、圧縮対象のデータ中から抽出した各マクロブロックをコードブック内の限られたコードベクトルの何れかによって代表させるも

のであるため、圧縮対象のデータ中で互いに類似した部分については、同じコード番号列が出力されることが多い。従って、ベクトル量子化後のコード番号列は、同じフレーズのデータ列が多く出力されることが想定されるため、このコード番号列に対してLZSS辞書圧縮を行った場合に、辞書として記憶されるデータ列を多数得ることができる。従って、ベクトル量子化後のデータ列とLZSS辞書圧縮は親和性が高く、圧縮率の高いデータ圧縮を行うことができる。

【0060】以上のようなLZSS辞書圧縮は完全可逆圧縮であるので、再生時にはLZSS辞書圧縮前のベクトル量子化によるコード番号列に戻すことができる。

【0061】次に、図7を参照しながら本実施形態のデータ圧縮方法の処理について説明する。図7は、上述の処理によるデータ圧縮方法の手順を示すフローチャートである。

【0062】先ず、ステップS1では、入力画像の入力が成される。そして、ステップS2では、所定領域5のR信号、G信号、B信号のそれぞれについてラプラシアンフィルタ6を用いてラプラシアン処理を行う。

【0063】次のステップS3では、ラプラシアン処理により得られた値 $L_R$ 、 $L_G$ 、 $L_B$ から、最大値を算出する。そして、所定領域5の全ての画素について最大値を求めてデータ群 $L_{MAX}$ を得る。そして、ステップS4、S5において、 $L_{MAX}$ としきい値30、60との比較を行う。

【0064】ステップS4において、 $L_{MAX}$ のうち少なくとも1つの値がしきい値60よりも大きい場合には、ステップS6へ進み縮小率を $1/1$ に設定してステップS10で色変換処理を行う。

【0065】ステップS4で $L_{MAX}$ が60未満である場合には、ステップS5へ進み、 $L_{MAX}$ としきい値30との比較を行う。ステップS5の比較により、 $L_{MAX}$ が30以上の場合には、ステップS7へ進んで縮小率を $1/2$ に設定してステップS9で解像度変換処理を行う。

【0066】ステップS5で $L_{MAX}$ が30未満である場合には、ステップS8へ進んで縮小率を $1/4$ に設定してステップS9で解像度変換処理を行う。

【0067】ステップS9で解像度変換処理を行った後、RGBデータから輝度、色差信号への変換を行い（ステップS10）、ベクトル量子化のためのマクロブロック化を行い（ステップS11）、ベクトル量子化を行う（ステップS12）。その後、ステップS13では、出力されたコード番号のDPCM処理を行い、ステップS14でLZSS処理を行う。LZSS処理後の圧縮データは、ステップS15で記憶する。

【0068】そして、エッジの大小に応じてマクロブロックの大きさを可変することにより、特にエッジが少なく検出された領域の画像の圧縮率を向上させることが可能となる。

【0069】次に、図8を参照しながら、上述のデータ圧縮方法を実現するデータ圧縮装置の構成について説明する。図8は、本実施形態のデータ圧縮装置の構成を示すブロック図である。

【0070】先ず、原画像である入力画像は画像入力部101へ入力される。そして、所定領域5における各画素のR信号、G信号、B信号が102においてラプラシアン変換される。各画素においてラプラシアン変換されたR信号、G信号、B信号のそれぞれ（ $L_R$ 、 $L_G$ 、 $L_B$ ）は、最大値検出部103において比較され、各画素毎に最大値の抽出が行われる。これにより、所定領域の各画素における最大値を抽出したデータ群 $L_{MAX}$ が抽出される。

【0071】最大値のデータ群 $L_{MAX}$ は、比較部104において予め定めた所定のしきい値（＝30、60）と比較される。そして、しきい値との比較の結果、縮小率決定部105において縮小率が決定される。

【0072】解像度変換処理部111では、決定された縮小率に基づいて所定領域5の解像度変換処理を行う。色変換処理部112においてRGBデータから輝度、色差信号への変換を行った後、所定領域5毎にマクロブロック化部106での処理を行い、ベクトル量子化部107においてベクトル量子化を行う。

【0073】DPCM処理部108は、ベクトル量子化部107から出力されたコード番号データに対して差分予測符号化処理を行う。そして、LZSS処理部109は、DPCM処理部108から出力されるコード列をLZSS処理により更に圧縮する。圧縮データ記憶部110は、圧縮したデータを記憶する。

【0074】次に、図10を参照しながら、本実施形態によるデータ圧縮を行った後の圧縮データについてのデータ伸長方法について説明する。

【0075】先ず、ステップS21では、上述の方法により生成された圧縮データを入力する。続くステップS22では、LZSS辞書伸長を行う。この処理は、LZSS辞書圧縮の逆変換である。

【0076】次に、ステップS23では、DPCM伸長を行う。この処理はDPCM処理の逆変換である。次に、ステップS24では、ベクトル量子化伸長処理を行い、次のステップS25で逆マクロブロック変換を行い、ステップS26において輝度、色差信号からRGBデータへの変換を行う。

【0077】次のステップS27では、解像度変換処理の逆変換を行う。これにより、解像度変換処理を行った所定領域については、縮小率に応じて拡張する処理が行われる。これにより、各所定領域毎に再生画像が生成される。再現された画像は、ステップS28で出力される。

【0078】以上説明したように、本実施形態によれば、入力画像のコントラストが低い場合であっても、R



信号、G信号、B信号のそれぞれでラプラシアン処理を行い、それぞれについて最大値を検出することにより、画像全体のコントラストが低い場合であっても確実にエッジ検出を行うことが可能となる。

【0079】従って、特に活字データの電子ファイル、カラーFAXアプリケーションに適用することで、色彩がついた活字であっても背景画像とのコントラストが低い場合であっても、確実にエッジ判定を行うことができる。また、コントラストの低いイメージ画像に対しても、エッジ判定を確実に行うことができる。

【0080】そして、エッジの大小に応じて線形補間処理を行うことにより、特にエッジが少ない領域の画像の圧縮率を向上させることが可能となり、ベクトル量子化の前段階での圧縮率を向上させることができる。

【0081】（その他の実施形態）上記様々な実施形態に示した各機能ブロックおよび処理手順は、ハードウェアにより構成しても良いし、CPUあるいはMPU、ROMおよびRAM等からなるマイクロコンピュータシステムによって構成し、その動作をROMやRAMに格納された作業プログラムに従って実現するようにしても良い。また、上記各機能ブロックの機能を実現するように当該機能を実現するためのソフトウェアのプログラムをRAMに供給し、そのプログラムに従って上記各機能ブロックを動作させることによって実施したものも、本発明の範疇に含まれる。

【0082】この場合、上記ソフトウェアのプログラム自体が上述した各実施形態の機能を実現することになり、そのプログラム自体、及びそのプログラムをコンピュータに供給するための手段、例えばかかるプログラムを格納した記録媒体は本発明を。かかるプログラムを記憶する記憶媒体としては、上記ROMやRAMの他に、例えばフロッピー（登録商標）ディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-I、CD-R、CD-RW、DVD、zip、磁気テープ、あるいは不揮発性のメモリカード等を用いることができる。

【0083】また、コンピュータが供給されたプログラムを実行することにより、上述の実施形態の機能が実現されるだけでなく、そのプログラムがコンピュータにおいて稼働しているOS（オペレーティングシステム）あるいは他のアプリケーションソフト等の共同して上述の実施形態の機能が実現される場合にもかかるプログラムは本発明の実施形態に含まれることは言うまでもない。

【0084】さらに、供給されたプログラムがコンピュータの機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに格納された後、そのプログラムの指示に基づいてその機能拡張ボードや機能拡張ユニットに備わるCPU等が実際の処理の一部または全部を行い、その処理によって上述した実施形態の機能が実現される場合にも本発明に含まれることは言うまでも

ない。

【0085】

【発明の効果】本発明によれば、コントラストが低い部分がある画像であっても、画像のエッジを確実に検出してエッジ量に応じた圧縮を行うことが可能となる。従って、再生画像の画質を維持したまま、更なる圧縮率の向上を達成することが可能となる。

【図面の簡単な説明】

【図1】本発明の一実施形態におけるエッジ検出の方法を示す模式図である。

【図2】線形補間処理によるデータ圧縮の方法を示す模式図である。

【図3】線形補間処理によるデータ圧縮の具体例を示す模式図である。

【図4】ベクトル量子化後の出力コード同士の差の分布を示す特性図である。

【図5】差分予測符号化処理により得られたコード列を示す模式図である。

【図6】ベクトル量子化により出力されたコード番号、コード同士の差の値、DPCM処理後のデータの関係を示す模式図である。

【図7】本発明の一実施形態におけるデータ圧縮方法の処理手順を示すフローチャートである。

【図8】本発明の一実施形態におけるデータ圧縮装置の構成を示す模式図である。

【図9】LZSS辞書圧縮の方法を示す模式図である。

【図10】本発明の一実施形態におけるデータ伸長方法の処理手順を示すフローチャートである。

【図11】本発明の一実施形態におけるベクトル量子化による画像圧縮を説明するための模式図である。

【符号の説明】

1 原画像（入力画像）

2 入力画像ブロック

3 コードブック

4 再現画像

5 所定領域

6 ラプラシアンフィルタ

11～14 画素

15 （線形補間処理により代表された）画素

101 画像入力部

102 ラプラシアン処理部

103 最大値検出部

104 比較部

105 縮小率決定部

106 マクロブロック化部

107 ベクトル量子化部

108 DPCM処理部

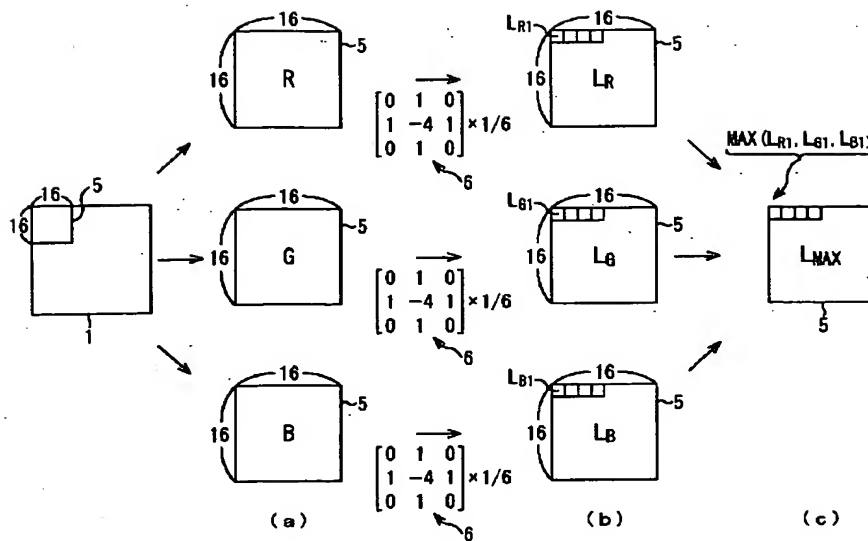
109 LZSS処理部

110 圧縮データ記憶部

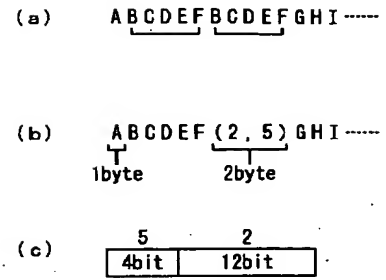
50 111 線形補間処理部



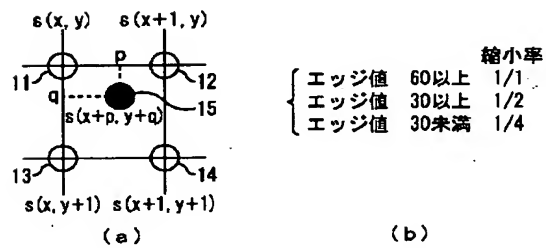
【図1】



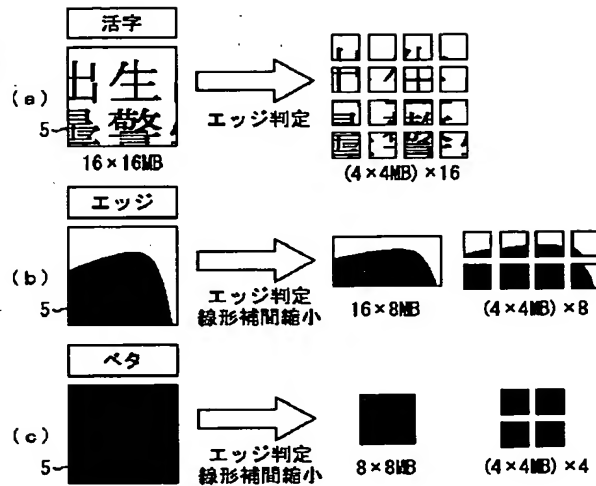
【図9】



【図2】



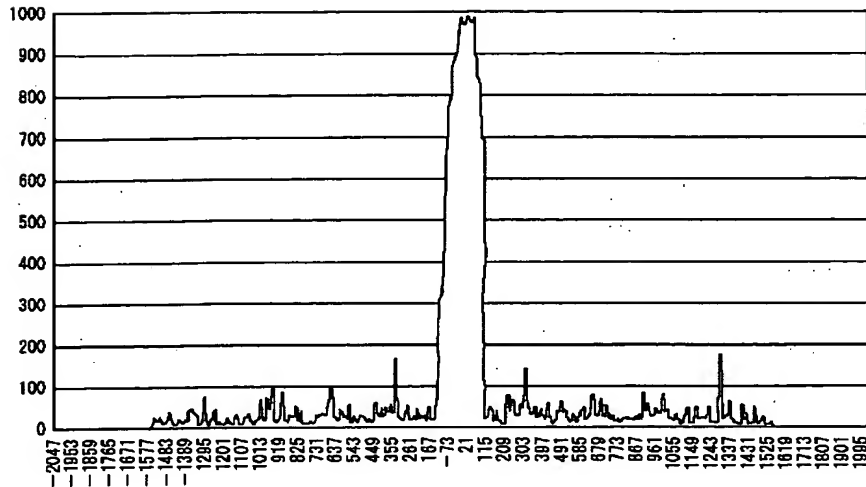
【図3】



【図6】

|             |    |       |      |            |        |
|-------------|----|-------|------|------------|--------|
| コード番号       | 2  | 3     | 1024 | 1023       | 5..... |
| 差           | -1 | -1021 | +1   | +1018..... |        |
| DPCM処理後のデータ | 2  | -1    | 1024 | +1         | 5..... |

【図4】

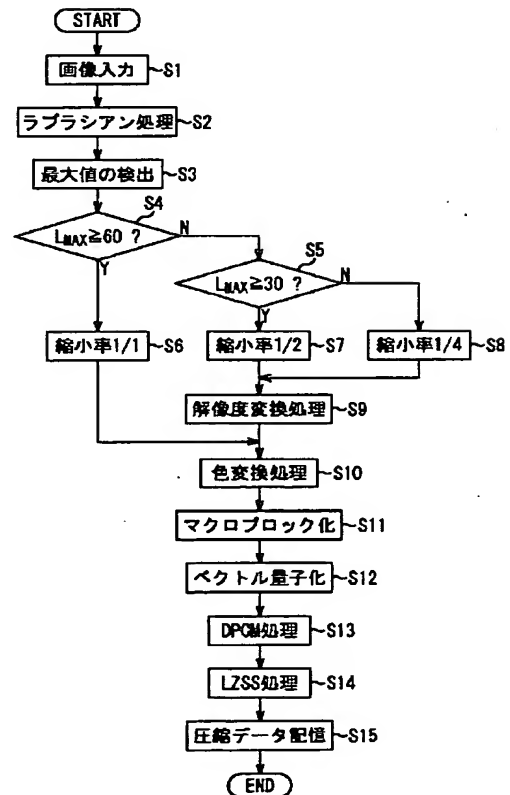


【図5】

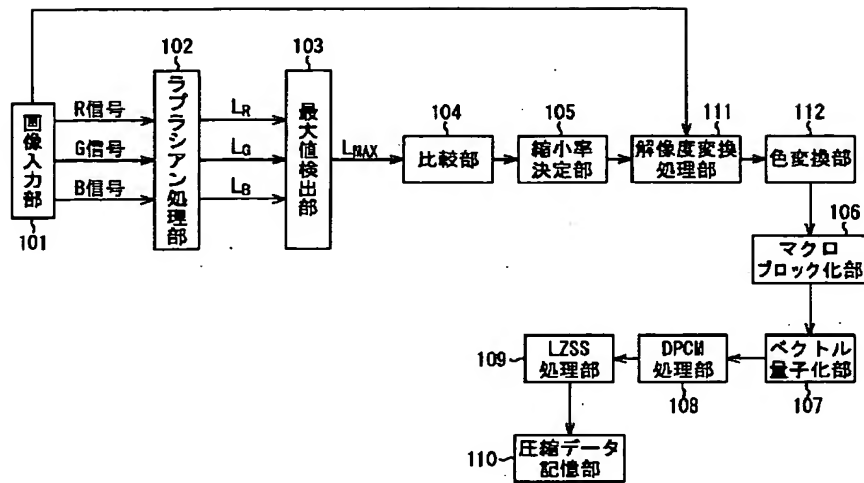
| 差分   | 出力コード      | 出力ビット数 | 処理   |
|------|------------|--------|--|
| -1   | 0xF0001コード | 16bit  | $\Delta\text{Code}=\text{Code}-\text{PreCode}$<br><br>if ( $\Delta\text{Code}\geq 0$ )<br>then $d=-\Delta\text{Code}*2$ ;<br>else $d=-\Delta\text{Code}*2-1$ ;<br><br>if ( $d\leq 0xF0$ )<br>then Write4bit( $d$ );<br>else Write16bit( $0xF0001\text{ Code}$ );<br><br>$\text{PreCode}=\text{Code}$ ; |
| -119 | 0xEF       | 8bit   |  |
| :    | :          |        |  |
| -5   | 9          |        |  |
| -4   | 7          |        |  |
| -3   | 5          |        |  |
| -2   | 3          |        |  |
| -1   | 1          |        |  |
| 0    | 0          |        |  |
| 1    | 2          |        |  |
| 2    | 4          |        |  |
| 3    | 6          |        |  |
| 4    | 8          |        |  |
| 5    | 10         |        |  |
| :    | :          |        |  |
| 119  | 0xEF       |        |  |
| 8    | 0xF0001コード | 16bit  |  |

出力コード番号列と直前のコード番号との差分を出し  
 ±119の場合は右に示す変換を行って8bitデータを出力する  
 ±119の範囲外のときは0xF000とコード番号との論理和(OR)  
 を取った16bitデータを出力する

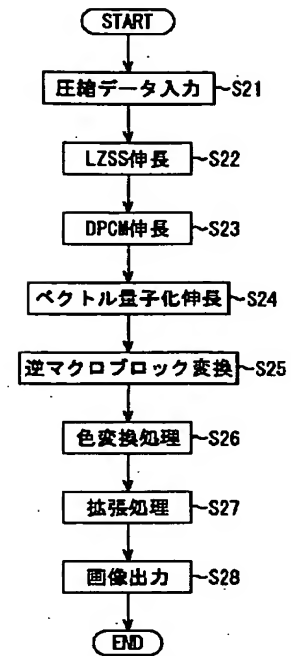
【図7】



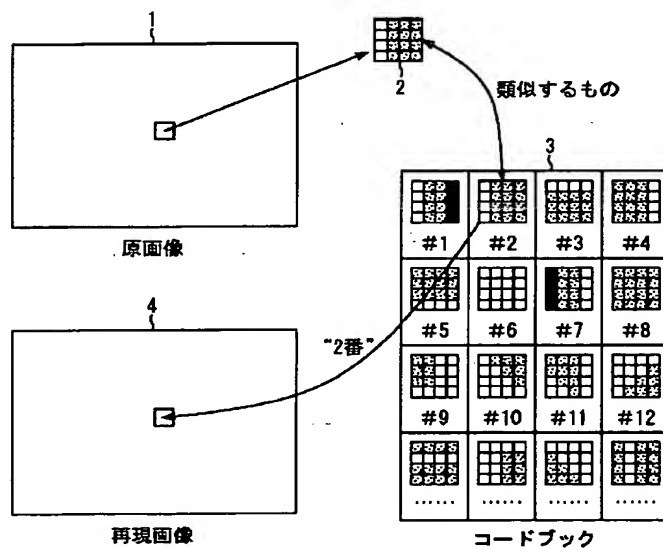
【図8】



【図10】



【図11】



フロントページの続き

(72)発明者 譽田 正宏  
東京都文京区本郷4丁目1番4号 コスモ  
ス本郷ビル アイ・アンド・エフ株式会社  
内

Fターム(参考) 5C059 KK18 LB11 MA37 MC18 MC35  
MD00 MD07 PP15 PP20 SS20  
UA02 UA11 UA39  
5C078 AA09 BA35 BA62 CA01 DA00  
DB00 DB04  
5J064 AA02 BA13 BA15 BB01 BB03  
BC12 BC14 BC22 BD03